Volume 4 Issue IV, April 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Secured Retrieval of Data with Resource Constrained Devices Using OPoR

Sakthi Sree T[#1], Aishwarya V[*2], GowthamKumar T[#3], Safia Ahmadha S[*4]
[#1]Assistant Professor, Department of Information Technology
[#2,3,4] B.Tech – Information Technology, Anna University, Chennai

Abstract — Cloud storage server is used to store the end user applications and their data's in a centralized storage area but the cloud service provider itself is not trustworthy. In this paper, we are going to ensure the three main characteristics like data integrity, dynamic data operations as well as the public verifiability by using a new scheme called Proof of Retrievability (PoR). To reduce the overheads and the computational cost due to the tag generation for files at the user side, public verifiability scheme is used. To tackle all these challenges, our new scheme has introduced two servers namely cloud storage server and cloud audit server. The main task of the cloud audit server is to preprocess the data and check for integrity on behalf of the cloud users. It also eliminates the user involvement in the tag generation in preprocessing and auditing phases. In addition to that, PoR model also provides defense against the problems like spoofed reset attacks that is caused by the cloud storage server in the upload phase.

Keywords — Data Integrity, Dynamic Data Operations, Public Verifiability, Proof of Retrievability, Reset Attacks

I. INTRODUCTION

Cloud computing and storage solutions are used by customers and enterprises to store and process their data in third-party data centers. It is based on resource sharing to achieve coherence and economies of scale, similar to a utility (like the electricity grid) over a network. The advantages of using cloud computing is that it reduces infrastructure costs and helps enterprises to concentrate on different projects and also allows the applications to run faster with less maintenance at the user side. It also has additional advantages such as high computing power, cheap cost of services, high performance and it is highly scalable, highly accessible as well as available. In the simplest terms, cloud computing means internet is used for storing the data and accessing it later instead of your computer's hard drive. For example, If you are an organization, and you want to use an online invoicing service instead of updating the in-house one you are using currently, the service which you use from internet is a "cloud computing" service. Rather than storing the data on your own hard drive and updating it later, you can use a service over the Internet, at another location, for storing your information and use the applications it provides. Doing so may give rise to certain privacy implications. In order to support the above processes cloud computing uses large group of servers which contains large pools of systems that are linked together. Cloud computing uses the "pay as you go model". The companies can thus increase the amount of computing when the need increases and decrease the computing, when demand decreases. Although, cloud computing has several advantages cloud service provider (CSP) cannot be trusted for integrity of the data that is stored. One of the major failures that may occur during this process is byzantine failure which is hidden by the service provider to the clients.

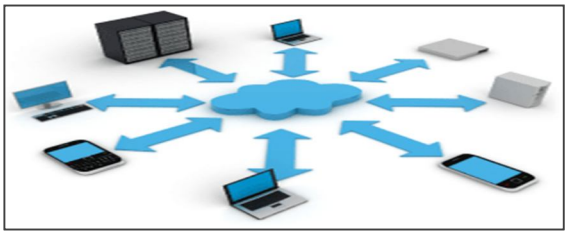

Fig 1: Various Devices Connected Via Cloud

Byzantine failures are random deviations of a process from its assumed behavior based on the algorithm it is supposed to be running and the received inputs. Such ***failures*** can occur, e.g., due to program error, hardware defect. Similarly, to save money and

Volume 4 Issue IV, April 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

memory the cloud service provider might dispose the rarely accessed client data's. To overcome such problems many people proposed various schemes. There are two main schemes for integrity check available based on the verifier role – private verifiability, public verifiability. The client system may crash when there are frequent integrity checks. Private verifiability is highly effective but it makes computation at the user side, a burden. Public verifiability is thus recommended as it reduces the computation burden for the users by authorizing a third party to perform verification. There is another serious problem of dynamic data support, that is the cloud storage previously did not support the modification of the data stored (i.e.,) adding additional data, removing existing data and updating existing data etc. Here, we have proposed schemes for combining both public verifiability and dynamic data storage. We propose a scheme called OPoR which contains two different servers. One is the cloud audit server (CAS) and the other one is cloud storage server (CSS). The cloud audit server is responsible for preprocessing the data before it is moved to CSS. The CAS is responsible for tag generation and it rescues computational burden on the users. The CAS is not required to have large storage space. The scheme we proposed supports both dynamic data operations and also provides security for reset attacks.

II. CHARACTERISTICS

The following are the characteristics of cloud computing: on-demand self-service, broad network access, resource pooling, rapid fluidity and quantified service. On-demand self-service means the customer sends request to the cloud service provider and the request is managed and processed by the service provider. In broad network access the services are offered to the customers via private networks or internet. In Pooled resources the customers draw resources from a pool of resources, usually in remote centers. Services can be computed larger or smaller; and customers are billed accordingly.

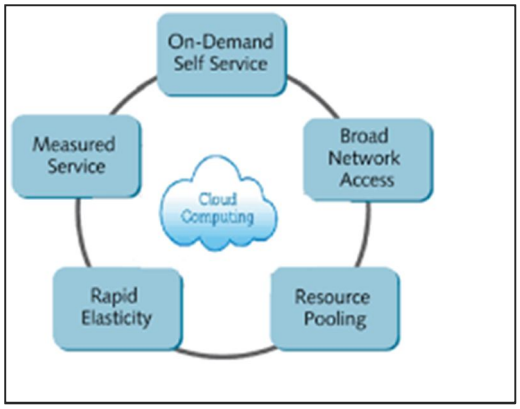

Fig 2: Characteristics of Cloud

III. SERVICE MODELS

There are three service models in cloud they are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). In a SaaS model, a readily-made application, with any necessary software, OS, network and peripherals are provided. In Platform as a Service, an OS, hardware, and network are provided, and the customer can install or develop his own set of software's and applications. The Infrastructure as a Service model provides only hardware and network; the customer can install and develop his own operating systems, software and applications.

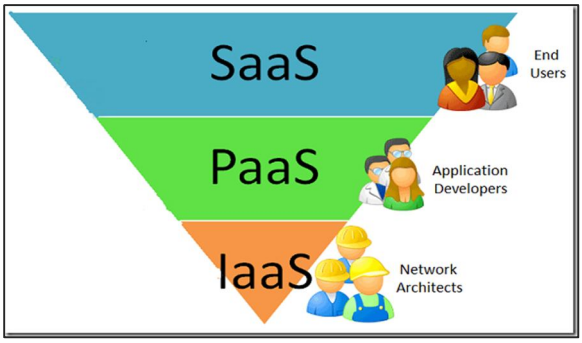

Fig 3: Cloud Service Models

Volume 4 Issue IV, April 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

IV. RELATED WORK

Many people have done research on security, integrity and dynamic data operations on the data that is stored in cloud. Ateniese et al. [1] defined the "provable data possession" (PDP) ensures data possession on untrusted data centers. They introduced schemes for public verifiability and the tag generation is based on RSA algorithm. However it does not support dynamic data operations for which they proposed dynamic version of PDP but even then it supported only dynamic data operations partially. Baochun Li[2] proposed that as the cloud data is shared among multiple users the data is under the doubt of being corrupted and thus states that public verification of such data will reveal the identity privacy to public verifiers. Thus he proposes a novel privacy-preserving mechanism that supports public auditing on shared data stored in the cloud using ring signatures to compute verification of metadata needed to audit the correctness of shared data. With this, the uniqueness of the signatory on each block in shared data is kept private from public verifiers to prevent privacy. It also processes multiple auditing tasks simultaneously. Y. Zhu, H. Hu, G.-J. Ahn, and M. Yu [10] proposed a new CPDP scheme based on multiprover zero knowledge proof system which satisfies completeness and zero-knowledge properties. It also minimizes the computational costs of the clients and service providers. In our study, we improve these features further and propose an efficient verification scheme where it supports both public verifiability and dynamic data operations. The tags are generated by the audit server and not by the user therefore the computational burden at the user is reduced. It also provides security against the reset attacks and provides efficiency analysis.

V. ARCHITECTURE

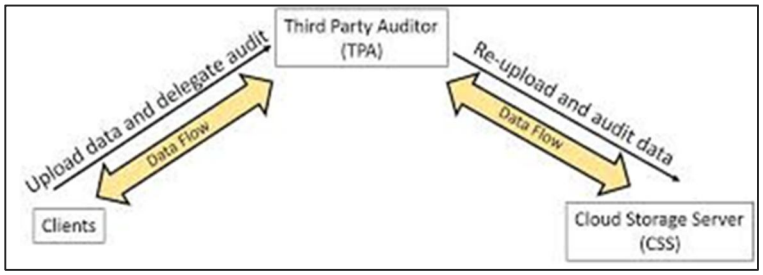

Fig 4: Cloud Storage Architecture

There are three entities involved in the cloud data storage process as follows:

A. Client

Client is the one who depends cloud storage server for storing the data and its maintenance.

Fig 5: Cloud Client

B. Third Party Auditor (TPA)

The TPA generates tags for file blocks before they are stored in cloud storage server and it reduces the computational burden of the clients. The TPA is capable of identifying the risks at CSP.

Volume 4 Issue IV, April 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

C. Cloud Storage Server (CSS)

The CSS has storage space required to store the client's data and the resources that are required to manage them. The cloud storage server is deployed by the cloud service provider. The cloud storage server provides proofs to the client and the TPA at the time of verification.

Fig 6: Cloud Storage Server

VI. DESCRIPTON

The client relies on the cloud storage server for data storage and maintenance and thus it does not locally possess a copy of the data that is outsourced to the CSS. In such case the integrity of the stored data cannot be checked without local possession of the data. So the client authorizes a third party auditor who in turn generates the tags for future verification of integrity of the data being outsourced.

VII. EXISTING SYSTEM

The existing scheme can simultaneously provide confirmable security in the upgraded security model and enjoy desirable efficiency, that is, no scheme can prevent reset attacks while backing efficient public verifiability and dynamic data operations simultaneously PoR model is the first to underpin dynamic update operations and guard against reset attack in a verification scheme. The robustness against reset attack ensures that a duplicate storage server can at no time gain any advantage of passing the verification of a falsely saved file by resetting the client (or the audit server) in the upload phase. We will see that most of existing PoR schemes cannot ensure this strong security for cloud storage.

VIII. PROPOSED SYSTEM

We present an efficient authentication scheme for providing distant data righteousness in cloud storage. The suggested scheme is proved to provide guard against reset attacks in the toughened security model while supporting efficient public verifiability and dynamic data transactions concurrently proposed a dynamic version of the prior PDP scheme. However, the system imposes a priori bound on the number of questions and do not fully support dynamic data operations. Wang et al. considered rapid data storage in shared scenario, and the proposed challenge-response protocol can both determine both data correctness and locate possible errors. Similarly they only considered partial support for dynamic data operation. Later, they also considered the methods to save storage space by announcing deduplication in cloud storage. Recently, Zhu et al introduced the provable data possession problem in a cooperative cloud service provider and designed a new remote integrity checking system.

IX. DESIGN GOALS

The design goals in our system can be summarized as the follows: (1) Public verifiability: to allow anyone, not just the clients originally stored the file, to have the capability to verify correctness of the remotely stored data; (2) Low computation overhead at the consumer side: to transmit data to the cloud server while supporting verifiability, the data owner does not have heavy additional computation; (3) Dynamic data operation support: to allow the clients to perform block level computations on the data files while sustaining the similar level of data correctness assurance; (4) Stateless verification: to discard the requirement for state information

Volume 4 Issue IV, April 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

retainment at the verifier side between audits and throughout the long term of data storage. This is also the basic requirement for achieving public verifiability. In particular, we aim to achieve enhanced security against reset attacks in our construction.

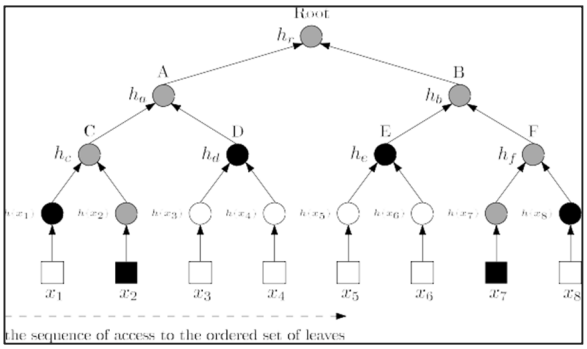

Fig 7: Merkle hash tree authentication of data elements. The access sequence of leaf nodes h(x1). . …., h(xn) is defined as the search order from left to right with depth first priority.

X. CORE CONSTRUCTION

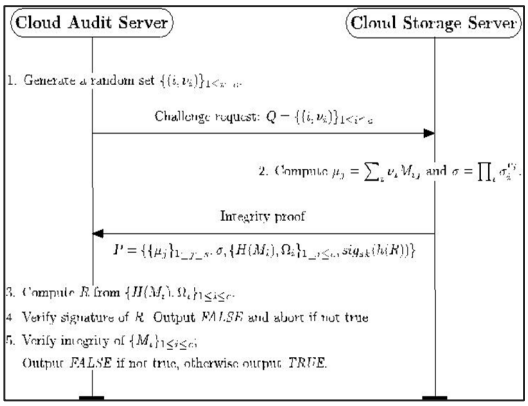

Fig 8: Protocols for integrity verification

The main objective of PoR model is to attain proof of retrievability. Informally, this property ensures that if an adversary can produce trusted integrity proofs of any record or file F for a nonnegligible section of challenges, for extracting F with overpowering probability, we can construct a PPT machine. It is suitably determined by the following game between a challenger C and a competitor A, where the role of audit server is played by C (the client) and A plays the role of the storage server:

A. Setup Phase

The setup algorithm for generating the key pair (pk, sk) is run by the challenger C and then forwards pk to the adversary A.

B. Upload Phase

C launches an empty dataless table called Rlist. A can query an upload oracle adaptively with reset capability as follows: – Upload: When a query appears from the user on a file F with a state index i, C checks if there is an entry (i, ri) in the R-list. If the result of the checking is yes, C overwrites the key ri onto its random tape; otherwise, C inserts (i, ri) into R-list where the content on its random tape is ri. Then C executes (F∗, t) ← Upload (sk, F; ri), and returns back the stored file F∗ and tag of the file tag t. The execution of the upload algorithm using randomness ri is denoted by (• ; ri).

C. Challenge Phase

The following two kinds of oracle queries can be adaptively made by A – Integrity Verify: When a query on a file tag t appears, C executes the integrity verification protocol and verifies the integrity for {A C (pk, t)} with A. – Update: When a query on a file tag t comes with a data operation request “update”, C executes the update protocol Update {A C (sk, t, update)} with A.

Volume 4 Issue IV, April 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

XI. CONCLUSION

In this study we propose a new scheme for proof of retrievability called OPoR for cloud storage which introduces a trusted third party audit server for preprocessing and uploading the data on behalf of the clients. Therefore the computational burden on the client side is significantly reduced. The audit server also performs data integrity verification and updation of the already outsourced data upon the client's request. Besides, we have constructed another scheme called PoR which is proved to be secure with enhanced security against the reset attacks in upload phase under PoR model. It also simultaneously supports the dynamic data operations and public verifiability.

XII. FUTURE WORK

The research line also includes various interesting topics. For instance, the trust on the cloud audit server can be reduced for more generic applications and the security against the reset attacks in data integrity verification protocol can be strengthened and efficient constructions for less storage and communication cost can be found. We leave the study of these problems as our future work.

REFERENCES

[1] Jin Li, Xiao Tan, Xiaofeng Chen, Duncan S. Wong, and Fatos Xhafa, "OPoR: Enabling Proof of Retrievability in Cloud Computing with Resource-Constrained Devices," IEEE Trans. Cloud Comput., vol. 3, no. 2, pp. 195-204,Apr-Jun. 2015.
[2] H. Li, B. Wang, and B. Li, "Oruta: Privacy-preserving public auditing for shared data in the cloud," IEEE Trans. Cloud Comput., vol. 2, no. 1, pp. 43–56, Jan.–Mar. 2014.
[3] H. Xiong, X. Zhang, D. Yao, X. Wu, and Y. Wen, "Towards end-to-end secure content storage and delivery with public cloud," in Proc. ACM Conf. Data Appl. Security Privacy, 2012, pp. 257–266.
[4] J. Li, X. Chen, J. Li, C. Jia, J. Ma, and W. Lou, "Fine-grained access control system based on attribute-based encryption," in Proc. Eur.Symp. Res. Comput. Security, 2013, pp. 592–609.
[5] J. Li, C. Jia, J. Li, and X. Chen, "Outsourcing encryption of attribute-based encryption with mapreduce," in Proc. 14th Int. Conf.Inf. Commun. Security, 2012, pp. 191–201.
[6] J. Li, X. Tan, X. Chen, and D. S. Wong, "An efficient proof of retrievability with public auditing in cloud computing," in Proc.5th Int. Conf. Intell. Netw. Collaborative Syst., 2013, pp. 93–98.
[7] Q. Zheng, and S. Xu, "Secure and efficient proof of storage with deduplication," in Proc. ACM Conf. Data Appl. Security Privacy, 2012, pp. 1–12.
[8] X. Chen, J. Li, J. Ma, Q. Tang, and W. Lou, "New algorithms of outsourcing modular exponentiations," in Proc. Eur. Symp. Res.Comput. Security, 2012, pp. 541–556.
[9] X. Lei, X. Liao, T. Huang, H. Li, and C. Hu, "Outsourcing large matrix inversion computation to a public cloud," IEEE Trans. Cloud Comput., vol. 1, no. 1, p. 1, Jan.–Jun. 2013.
[10] Y. Zhu, H. Hu, G.-J. Ahn, and M. Yu, "Cooperative provable data possession for integrity verification in multicloud storage," IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 12, pp. 2231–2244, Dec.2012.
[11] A. Juels and B. S. Kaliski Jr, "Pors: Proofs of retrievability for large files," in Proc. 14th ACM Conf. Comput. Commun. Security, 2007, 584–597.
[12] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability: Theory and implementation," in Proc. ACM Workshop Cloud Com-put. Security, 2009, pp. 43–54.
[13] M. Naor and G. N. Rothblum, "The complexity of online memory checking," J. ACM, vol. 56, no. 1, pp. 2:1–2:46, Feb. 2009.
[14] E.-C. Chang and J. Xu, "Remote integrity check with dishonest storage server," in Proc. 13th Eur. Symp. Res. Comput. Security, 2008, pp. 223–237.
[15] M. A. Shah, R. Swaminathan, and M. Baker. (2008). "Privacy-pre-serving audit and extraction of digital contents," Cryptology ePrint Archive, Report 2008/186 [Online]. Available: http://eprint.iacr. org/